



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/064,011

06/04/2002

Timothy G. Deboer

CA920010048US1

7212

68168 7590 06/18/2008  
MICHAEL BUCHENHORNER, P.A.  
8540 SW 83 STREET  
SUITE 100  
MIAMI, FL 33143

EXAMINER

BROPHY, MATTHEW J

ART UNIT

PAPER NUMBER

2191

NOTIFICATION DATE

DELIVERY MODE

06/18/2008

ELECTRONIC

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

michael@buchenhorner.com  
ana@buchenhorner.com

<b>Office Action Summary</b>	<b>Application No.</b> 10/064,011	<b>Applicant(s)</b> DEBOER ET AL.	
	<b>Examiner</b> MATTHEW J. BROPHY	<b>Art Unit</b> 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 11 April 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 16-19 and 26-37 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☐ Claim(s) 16-19 and 26-37 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

## DETAILED ACTION

1. This office action is in response to Request for Continued Examination filed April 11, 2008.
2. Claims 16-19 and 26-37 are now pending.

### *Response to Amendment*

### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 16-19 and 26-37 are rejected under 35 U.S.C. 102(e) as being anticipated by *Arnaiz et al* USPN 7,080,371.

Regarding these claims, Arnaiz teaches:

Regarding Claim 16 (Currently amended) Arnaiz teaches: A method of versioning server configuration files, said method comprising steps of: associating a server configuration with an application (**e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site,**

**and merge the customer's repository with a new repository.”);** storing multiple versions of both application and server configuration data required to run on a plurality of servers; selecting a single version of said application and the server configuration data **(e.g. Column 7, Lines 35-52, “Version Method. A specification of how to identify the current version of installed software component on a machine. For example, read the Windows registry to get the currently installed version of Microsoft Word on the local machine. "Compiled" component information. Information about the previously released version of the software component. For example Siebel Remote stores exactly two versions of software components. The user can view and manipulate the current version. Siebel Remote stores the previously released version in a LONG column in the software component table (i.e. S\_UPG\_COMP). The upgrader, e.g., the database administrator, uses the Component screen to copy the current version into the "compiled" LONG column. Schema. The schema is the database schema, also referred to as a meta-database. This includes the database tables, indexes, views, seed data and repository data that must be installed in the database to run. Client. The executables, dlls, reports, help files needed to run the client. CDF. The definition of a customer's configuration for the client. Server. The executables, dlls and other files needed to run the server. Upgrade wizard. The standalone executable that applies upgrade kits on machine.”);** within a package comprising an application, associating a version of a server configuration file with said application **(e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database**

**The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”), said version configuring a plurality of servers to process said application (e.g. Column 2, Line 3-8, “It is a further object of our invention to provide support for all types of installations and terminals in an enterprise. That is, the sol, are provider or the upgrader, e.g., the database administrators must be able to upgrade all types of the installations in an enterprise, including connected clients, remote clients, regional servers and application servers.”), and transferring the same server configuration data to each of the plurality of servers so that each server can execute the application solely on the server (e.g. Column 8, Lines 58-61, “After defining the upgrade, the upgrader, e.g., the database administrator, then initiates the upgrade from the server database. Client programs at all locations automatically detect and apply the upgrades as needed.”).**

Regarding Claim 17 (Previously Presented) Arnaiz teaches: The method of claim 16 further comprising a step of providing said associating with embedding association data within said version of said server configuration file linking said version of said server configuration file with said application **(column 11, lines 18-25, At the beginning of each synchronization session, the docking client checks whether a database extract or a database schema upgrade is pending. This is because the docking client must perform a schema version check at the start of the docking session because the docking client must know whether it can upload transactions to the server database or must discard the transactions. The**

**docking client also must know whether it must perform a dbinit to re-initialize the local database).**

Regarding Claim 18 (Previously Presented) Arnaiz teaches: The method of claim 16 further comprising a step of providing said associating with embedding association data with said application linking said application with at least one of said plurality of versions of said server configuration file **(column 11, lines 18-25, At the beginning of each synchronization session, the docking client checks whether a database extract or a database schema upgrade is pending. This is because the docking client must perform a schema version check at the start of the docking session because the docking client must know whether it can upload transactions to the server database or must discard the transactions. The docking client also must know whether it must perform a dbinit to re-initialize the local database).**

Regarding Claim 19 (Previously Presented) Arnaiz teaches: The method of claim 17 further comprising a step of providing said package further with said version of said server configuration file **((column 9, lines 26-36, Alternatively, the upgrader, e.g., the database administrator, can utilize an Upgrade Kit screen. The upgrader, e.g., the database administrator, uses the Upgrade Kit screen update the required versions of a software component. The upgrader, e.g., the database administrator, selects an upgrade kit, and, initiates a function to set the required versions for example, in the case of Siebel Remote, the upgrader presses the "Set**

**Required Versions" button. The button finds all the software components that the upgrade affects and sets their maximum versions. The maximum versions are set to the versions that the upgrade kit installs)**

Regarding Claim 26 (New) Arnaiz teaches: A method of concurrently updating application and server configuration data, said method comprising: providing to a server configuration versioning tool for storing in a data repository comprising multiple versions of both the application and the server configuration data required to run on a plurality of servers(e.g. Column 7, Lines 35-52, **"Version Method. A specification of how to identify the current version of installed software component on a machine. For example, read the Windows registry to get the currently installed version of Microsoft Word on the local machine. "Compiled" component information. Information about the previously released version of the software component. For example Siebel Remote stores exactly two versions of software components. The user can view and manipulate the current version. Siebel Remote stores the previously released version in a LONG column in the software component table (i.e. S\_UPG\_COMP). The upgrader, e.g., the database administrator, uses the Component screen to copy the current version into the "compiled" LONG column. Schema. The schema is the database schema, also referred to as a meta-database. This includes the database tables, indexes, views, seed data and repository data that must be installed in the database to run. Client. The executables, dlls, reports, help files needed to run the client. CDF. The definition**

**of a customer's configuration for the client. Server. The executables, dlls and other files needed to run the server. Upgrade wizard. The standalone executable that applies upgrade kits on machine.”):** at least one server configuration file comprising server configuration data and application association data (**e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”)** comprising a list of all applications installed on the server as well as versions of the server to which the applications are associated (**Column 2, Line 62-67, “After the table of contents of the software upgrade kits is installed, it is compared to the locally installed software on a client, with the Software needing to be installed on the client to the client being downloaded to the client. This comparison can be at the startup of the server, the client, or a particular software component.”)**; wherein the providing step is responsive to creation or modification of a server configuration file (**e.g. Column 8, Lines 58-61, “After defining the upgrade, the upgrader, e.g., the database administrator, then initiates the upgrade from the server database. Client programs at all locations automatically detect and apply the upgrades as needed.”)**; receiving a request to select one or more server configuration files stored within the data repository, the request comprising data associated with an application which is to be transferred to at least one server on at least one platform (**Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to**



**mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next synchronization session. This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”);** retrieving a latest version of the at least one server configuration file associated with the application to be transferred **(Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next synchronization session. This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”);** bundling the application and the latest version of the at least one server configuration file into a single package comprising data to configure the at least one server for running said application on the at least one platform **(Column 8, Line 6-8, “Each upgrade kit must contain all the files and commands needed to install the software components. For example, an upgrade kit can be used to install Microsoft Word 7.0.2 on all clients. This upgrade kit must contain all the files (e.g.**

**executables, sample files, templates, etc.)” and e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”); transferring to the at least one server the single package (Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next synchronization session. This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”); and configuring the at least one server with the server configuration data from the latest version of the server configuration file (e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”).**

Regarding Claims 27 and 34 (New) Arnaiz teaches: The method of claim 26 further comprising providing server configuration metadata for storing in the data

repository, the metadata comprising a linkage between the at least one server configuration file and corresponding application data (**e.g. Column 14, Line 63 to Column 15, Line 20, The Software Component applet lets the upgrader, e.g., the database administrator, enter the attributes of a software component. It can have one or more of the following fields: Name (required). The name of the software component. Comments (optional). The description of the software component. Minimum version (optional). The minimum version that must be installed on a machine to use this software component. If NULL, then the software component does not have a minimum version. Maximum version (optional). The maximum version that must be installed on a machine to use the software component. If NULL, then the software component does not have a maximum version. The applet verifies that the minimum version is less than or equal to the maximum version. Locate method (optional). The method to use to get the install location of the software component. This is a picklist. Location Info (optional). Additional information for the locate method. Version method (optional). Method to get the currently installed version of software component. This is a picklist. Version Info (optional). Additional information for the version method.**”).

Regarding Claims 28 and 35 (New) Arnaiz teaches: The method of claim 27 wherein the server configuration metadata is provided in the server configuration file (**Col. 16, Line 6-11, “Upgrade kit archive file (optional). The attributes for the attached upgrade kit archive file. These are the attributes used by the file**

**attachment business component and frames. Mobile users can navigate to this applet and request that Remote download the upgrade kit archive in the next synchronization session.”).**

Regarding Claims 29 and 36 (New) Arnaiz teaches: The method of claim 26 further comprising providing application metadata for storing in the data repository, the application data comprising at least one selected from a group consisting of: data describing a particular file or component, version data, timestamp data, and indicator data indicating whether a piece of data has been checked out and by whom **(e.g. Col. 15, Lines 4-9, “Maximum version (optional). The maximum version that must be installed on a machine to use the software component.”).**

Regarding Claim 30, 33 and 37 (New) Arnaiz teaches: The method of claim 26 further comprising a step of providing the at least one server configuration file with a first server configuration file adapted to configure a first server and further comprising a step of providing another of the at least one server configuration file with a second server configuration file adapted to configure a second server **(figures 1-3, column 5, lines illustrates one set of test and distribution procedures for customers. As shown in FIG. 3, upgrade CD's or diskettes (307) are created in a master repository (305) and shipped to a test database server (303) for distribution to test users, including server test users (313), connected test users (311), and mobile test users (315). The test users (311, 313, and 315) upgrade their test environments, and the upgraded version of the software for testing. The customers (311, 313,**

**315) test the software. If the test is satisfactory, the newly upgraded version of the software is distributed to production users (4121, 413, 415)).**

Regarding Claim 31 (New) Arnaiz teaches: The method of claim 26 further comprising a step of providing the at least one server configuration file with a first server configuration file adapted to configure a first server for execution on a first computer system platform and further comprising a step of providing another of the at least one server configuration file with a second server configuration file adapted to configure said first server for execution on a second computer system platform (**column 4, lines 40-55, FIG. 2 illustrates initiation and distribution of upgrades. In the process shown in FIG. 2 the administrator updates the required versions for a software item. This causes the database server, when started up, this causes the client to compare the required versions of the software with the locally installed versions for each software item. If an upgrade is required the client retrieves the upgrade kit archive file from the file server and invokes the upgrade wizard to apply the upgrade locally. Also, shown in FIG. 2 is upgrading of mobile users. In the case of mobile users an application server routes new required versions to mobile and regional users. After a docking session, the docking client compares the required versions to locally installed versions of the software. If an upgrade is required, the docking client retrieves the upgrade kit archive file from the file server and invokes the upgrade wizard to apply the upgrade the local machine).**

Regarding Claim 32: A computer storage medium storing data and instructions, said data and instructions adapting the computer system to: provide to a server configuration versioning tool for storing in a data repository comprising multiple versions of both application and server configuration data required to run on a plurality of servers **e.g. Column 7, Lines 35-52, "Version Method. A specification of how to identify the current version of installed software component on a machine. For example, read the Windows registry to get the currently installed version of Microsoft Word on the local machine. "Compiled" component information. Information about the previously released version of the software component. For example Siebel Remote stores exactly two versions of software components. The user can view and manipulate the current version. Siebel Remote stores the previously released version in a LONG column in the software component table (i.e. S\_UPG\_COMP). The upgrader, e.g., the database administrator, uses the Component screen to copy the current version into the "compiled" LONG column. Schema. The schema is the database schema, also referred to as a meta-database. This includes the database tables, indexes, views, seed data and repository data that must be installed in the database to run. Client. The executables, dlls, reports, help files needed to run the client. CDF. The definition of a customer's configuration for the client. Server. The executables, dlls and other files needed to run the server. Upgrade wizard. The standalone executable that applies upgrade kits on machine."**):

at least one server configuration file comprising server configuration data and

application association data comprising a list of all applications installed on the server as well as versions of the server to which the applications are associated (**e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”** And **Column 2, Line 62-67, “After the table of contents of the software upgrade kits is installed, it is compared to the locally installed software on a client, with the Software needing to be installed on the client to the client being downloaded to the client. This comparison can be at the startup of the server, the client, or a particular software component.”**); wherein the providing step is responsive to creation or modification of a server configuration file (**e.g. Column 8, Lines 58-61, “After defining the upgrade, the upgrader, e.g., the database administrator, then initiates the upgrade from the server database. Client programs at all locations automatically detect and apply the upgrades as needed.”**);

receive a request to select one or more server configuration files stored within the data repository, the request comprising data associated with an application which is to be transferred to at least one server on at least one platform (**Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next**

**synchronization session. This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”);**

retrieve a latest version of the at least one server configuration file associated with the application to be transferred **(Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next synchronization session.**

**This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”);**

bundle the application and the latest version of the at least one server configuration file into a single package comprising data to configure the at least one server for running said application on the at least one platform **(Column 8, Line 6-8, “Each upgrade kit must contain all the files and commands needed to install the software components. For example, an upgrade kit can be used to install Microsoft Word 7.0.2 on all clients. This upgrade kit must contain all the files (e.g. executables, sample files, templates, etc.)” and e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database**



**administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”);**

transfer to the at least one server the single package **(Column 8, Line 63 – Column 9, Line 6, “After the upgrade kit is defined, it is possible to distribute the upgrade kit definitions associated files to mobile and regional databases. In this case mobile users can navigate to the Upgrade Kit screen and request that the database management system download the upgrade kit archive file in the next synchronization session. This lets mobile users download large upgrade kit archive files long before the upgrade kit needs to be applied to the local machine. For example, the mobile user may be visiting headquarters and can download the upgrade kit archive file in much less time over the LAN than over a modem.”);** and configure the at least one server with the server configuration data from the latest version of the server configuration file **(e.g. Column 13, Lines 20-23, “Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.”).**

### ***Response to Arguments***

3. Applicant's arguments filed April 11, 2008 have been fully considered but they are not persuasive.

Art Unit: 2191

In remarks, Applicant Argues:

In Arnaiz, a "version upgrade kit" is created containing an upgrade wizard and software upgrades. Arnaiz's kit is distributed to clients by downloading them onto the client during docking with the server computer. The client computer then invokes the upgrade wizard.

Arnaiz, col. 2, lines 59-61: "Copies of the upgrade kits are downloaded to clients to be upgraded; and the software on the client is upgraded." Also see Arnaiz at col. 22, line 30 through col. 23, 7. Arnaiz does not provide for server configuration, much less synchronizing application versioning with server configuration to guarantee that the correct version of an application is always executed on a server properly configured to run that version of the application. Further, the Office Action did not provide the details required for an anticipation rejection. "A rejection for anticipation under section 102 requires that each and every limitation of the claimed invention be disclosed in a single prior art reference." *Karsten Manufacturing Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383 (Fed. Cir. 2001).

Examiner's Response:

Examiner respectfully disagrees. Arnaiz anticipates the "server configuration" of the claims at issue. For example, Column 13, Lines 8-29 teach the creation of server configuration files, the inclusion in the upgrade kit (herein interpreted to be equivalent to applicant's package).

Col. 13, Ln. 8-36,

"Customizing and Testing

Upgrader, e.g., the database administrator, can customize local installations in these three ways:

Database extensions such as new extension tables, extension columns or indexes. Configuration changes such as new CDF files, CFG files or report files.

Third party software changes such as new versions of Microsoft Word, Adobe Acrobat and the like.

Customers follow a well-defined procedure for creating and distributing these kinds of customizations. All of these customizations do not cause a major schema change; hence, they do not require a database re-extract.

Step 1: Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository.

To effect configuration changes during the version or schema upgrade, new .CDF files need to be generated. This is done using an editor to customize CFG files. And a DBMS program such as Microsoft Access to customize reports. Microsoft Help Editor or the like may be used to customize the Help files.

Programs in the upgrade kit compare versions and automatically detect the need to apply an upgrade the next time they start up. The method, apparatus, system,

and program product of our invention upgrades connected users. In a similar way, the server upgrades application servers, or, when present, a docking client upgrades local databases. A Replication agent upgrades regional databases.”

In remarks, Applicant Argues:

Arnaiz does not teach any of the steps of claim 26. Arnaiz neither teaches nor suggests a package comprising server configuration data comprising data to configure each of said servers for running said application. Arnaiz's "kit" is not analogous to the package of claim 26. The package contains the application to be executed, along with the corresponding version of the server configuration file or files necessary to correctly configure the server to run that version of the application. Arnaiz's kit, a software upgrade kit, contains a table and upgrade files and is executed by an upgrade wizard invoked by the server. See Arnaiz, col. 6, lines 39-40 and col. 5, lines 40-43. Arnaiz, in contrast to the claimed invention, does not synch user application versions with server configuration versions. Moreover, Arnaiz neither teaches nor suggests associating a server configuration with an application. The final Office Action contends that Arnaiz at col. 22- , "fairly suggest package comprising server configuration data." Not only does Arnaiz not suggest such a package, but Applicant respectfully reminds the Examiner that for purposes of an anticipation rejection the prior art must do more than suggest; it must enable the claimed feature. See Soundsciber, supra. Claim 1 is a method claim and in order to anticipate claim 1 a prior art reference would have to teach or enable the claimed method steps. The Office Action cites a computing system claim (claim 6 of

Art Unit: 2191

Arnaiz) for the limitation of : "transferring to each of said plurality of servers the single version of a package, said package comprising said application and server configuration data, said server configuration data comprising data to configure each of said servers for running said application." That cited portion is not legally sufficient to make a prima facie case of anticipation. *Los Alamitos Sugar Co. v. Carroll*, 173 F. 280 (9th Cir. 1909)(it is not sufficient to constitute anticipation that the devices relied upon might, by a process of modification, reorganization or modification with each other be made to accomplish the same function as the claims at issue).

Examiner's Response:

Examiner Respectfully disagrees. As described above, Arnaiz anticipates the server configuration of the claims at issues. In addition, the passage quoted above, teaches the synchronizing of the configuration of servers and the applications in the upgrade kit of Arnaiz. From the passage above: Col 13. "Configuration changes such as new CDF files, CFG files or report files...Make the configuration changes on the server database The upgrader, e.g., the database administrator, uses programs to customize end-user installations for their site, and merge the customer's repository with a new repository... To effect configuration changes during the version or schema upgrade, new .CDF files need to be generated. This is done using an editor to customize CFG files...

Programs in the upgrade kit compare versions and automatically detect the need to apply an upgrade the next time they start up. The method, apparatus, system, and program product of our invention upgrades connected users."

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MATTHEW J. BROPHY whose telephone number is 571-270-1642. The examiner can normally be reached on Monday-Thursday 8:00AM-5:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MJB

6/12/2008  
/Wei Zhen/  
Supervisory Patent Examiner, Art Unit 2191